GENERATION OF REPEATABLE CRYPTOGRAPHIC KEY BASED ON VARYING PARAMETERS

This application claims the benefit of U.S. Provisional Application Serial No. 60/128,413, filed April 8, 1999 and U.S. Provisional Application Serial No. 60/147,880 filed August 9, 1999, both of which are incorporated herein by reference.

Field of the Invention

The present invention relates generally to computer security. More particularly, the present invention relates to the generation and use of cryptographic keys for computer security applications.

Background of the Invention

As computer use becomes more widespread, the problem of computer system security also becomes increasingly critical. The volume of information stored in computer systems is growing at a large rate. Further, the accessibility of such information systems is increasing due to the interconnection of computer systems through networks such as the Internet. A major problem facing computer owners is how to protect computer systems, and the information they contain, from adversaries wishing to gain unauthorized access to stored information.

One type of computer system security is referred to as authentication. Authentication refers to confirming the identity of a user prior to allowing access to a computer system. Most authentication schemes are based on the user's knowledge of a secret, called a *password.* A user must have knowledge of a secret password in order to gain access to the computer system.

Another type of computer system security is referred to as encryption. Some, or all, of the information on the computer system may be encrypted such that the information is rendered unreadable or unusable until it is decrypted. Like authentication, decryption also relies on the knowledge of a secret, called a *key*, which is used to decrypt the information. Thus, even though a person may have access to information, that information may be useless to someone who does not possess the appropriate decryption key.

These two security techniques of authentication and encryption are related in many ways. For example, a secret known by a user may serve as both a password and a decryption key. Further, a computer system may employ both types of security techniques.

With respect to authentication, textual passwords have been, and remain, the primary means of authenticating users. However, passwords have been shown to be a relatively weak mechanism for authentication. Studies have shown that users tend to choose passwords that can be easily guessed by an exhaustive search of a relatively small subset of all possible passwords. For example, in a study of 14,000 computer system passwords, it was found that almost 24% of the passwords could be found in a "dictionary" of only 3×10^6 words. Considering the high speed at which a computer could generate and test 3×10^6 words, passwords are considered to be a weak form of computer security.

One known technique for strengthening passwords is to require not only that the correct password be typed, but also that the user's keystroke features (e.g. duration of keystrokes and latency between keystrokes) match a predetermined stored model of expected keystroke features. This technique is effective against so-called online attacks in which an adversary is attempting to gain access to a computer system through the computer's authentication system. However, this technique is not effective against a so-called off-line attack, in which an adversary gains physical access to the computer's data, for example by taking physical possession of a laptop computer or by otherwise circumventing the computer's authentication system. Once an adversary has physical access to computer information, the above described keystroke feature technique is ineffective. Further, if an adversary gets physical access to a computer which allows access to the stored keystroke feature models, the models may leak sensitive information which would then make it easier for the adversary to determine actual user passwords.

Other techniques exist which do not require the storage of such models in memory. For example, U.S. patent no. 5,680,460 describes a technique in which a user's fingerprint characteristics are measured and various filters are applied to the measurements to generate a key which can then be used to authenticate the user on a computer system. Another example is G.I. Davida, Y. Frankel, and B.J. Matt, *On Enabling Secure Applications Through Off-Line Biometric Identification*, Proceedings of the 1998 IEEE Symposium on Security and Privacy, pp. 148-157, May 1998, in which error correcting parameters are used to decode biometric (e.g. iris scan) readings into a canonical form for a particular user. This canonical form may then be used to generate a key for authentication purposes. However, both of these techniques also suffer from the above described deficiency in that any compromise of the underlying system data (either the filters or the error correcting parameters) will leak sensitive information which, in certain applications, would allow an adversary to more easily determine the user's authentication key.

Summary of the Invention

The present invention provides for the generation of a repeatable cryptographic key based on potentially varying parameters which are received, for example, during a computer resource access attempt. The key is repeatable in that the same key may be generated using different received parameters. In accordance with the invention, so-called cryptographic shares are retrieved from memory locations identified as a function of the parameters. The key may be determined from knowledge of a sufficient number of cryptographic shares.

In accordance with one embodiment, values of the received parameters are used to generate indices into a so-called share table which stores valid and invalid cryptographic shares. Valid cryptographic shares (i.e., those which may be used to generate the key) are stored in memory locations whose indices are expected to be generated during legitimate access attempts. Invalid cryptographic shares (i.e., those which may not be used to generate the key) are stored in memory locations whose indices are not expected to be identified during legitimate access attempts. Thus, the share table may be configured to take into account expected variations in the received parameters. In accordance with one technique, valid shares may be periodically replaced with invalid shares (and vice versa) based on a history of access attempts and a change in expected parameter values.

In various embodiments, the stored shares may be encrypted in various ways. In one embodiment, the shares are encrypted using a value computed as a function of expected parameter values. In this embodiment, one encrypted share is stored and associated with each parameter. When a parameter is received during an access attempt, the associated stored encrypted share is retrieved from memory and an attempt to decrypt the share is made using a value computed as a function of the parameter value. Only if the computed value is the same as the value used to encrypt the share (i.e., the expected value) will a valid share be obtained. Otherwise, an invalid share will be obtained. In various embodiments, the cryptographic shares may be chosen so as to allow generation of the key even if some number of invalid shares are generated, as long as a sufficient number of valid shares are generated. This embodiment is particularly advantageous when the number of possible parameter values is very large. This embodiment is described in further detail below in accordance with an embodiment of the invention in which the contents of a database may be decrypted using parameters representing measured DNA information.

Various secret sharing schemes may be used to generate the cryptographic shares. In accordance with one embodiment of the invention, a so-called polynomial secret sharing scheme is used. In this embodiment, the secret key is the point where the polynomial crosses the y axis and the cryptographic shares are points on the polynomial. Knowledge of a sufficient number of points on the polynomial (i.e., cryptographic shares), allows for the determination of the polynomial, and thus determination of the key. These polynomial points may be stored in a straightforward manner by storing (x,y) pair values in the share table. Alternatively, in order to save memory space (at the expense of an increase in computational complexity) compressed shares may be stored in the share table.

In accordance with an advantageous embodiment of the invention, the inventive techniques are used to implement a keystroke feature authentication system in which users are authenticated based on a combination of an entered password and the manner in which the password was typed. In accordance with this embodiment, a share table contains a row for each keystroke feature to be considered (e.g., duration of keypresses, latencies between keypresses). Upon receipt of a measured keystroke parameter, a function chooses one of two columns in the share table associated with the parameter row, depending on a comparison of the parameter with a predetermined threshold. A history of legitimate authentications is kept. When a measured keystroke parameter consistently results in the function choosing one of the two columns, the associated keystroke feature is considers a distinguishing feature, and the other of the two columns in the share table is updated to contain an invalid cryptographic share. Over time, the share table is updated such that only share table entries which are expected to be accessed during future legitimate access attempts contain valid cryptographic shares, while the share table entries which are not expected to be accessed during a legitimate access attempt contain invalid cryptographic shares. In this manner, an illegitimate access attempt will retrieve invalid cryptographic shares from the share table such that it will not be possible to generate the cryptographic key. In accordance with another aspect of this embodiment, the shares stored in the share table are encrypted using the password as the decryption key in order to add additional security.

These and other advantages of the invention will be apparent to those of ordinary skill in the art by reference to the following detailed description and the accompanying drawings.

Brief Description of the Drawings

Fig. 1 shows an exemplary share table;

Fig. 2 illustrates an exemplary polynomial which may be used in conjunction with a polynomial secret sharing scheme;

Fig. 3 illustrates a share table which stores compressed cryptographic shares;

Fig. 4 illustrates a share table which stores encrypted cryptographic shares;

Fig. 5 illustrates a share table in accordance with a keystroke features authentication embodiment of the invention; and

Fig. 6 illustrates a history table in accordance with a keystroke features authentication embodiment of the invention.

Detailed Description

The present invention provides for the generation of a repeatable cryptographic key based on a set of potentially varying parameters. The cryptographic key is repeatable in that the same key is generated notwithstanding that the parameters, on which generation of the key is based, may vary from one generation of the key to the next. In an advantageous embodiment, the parameters represent measurements of some physical characteristics of either a person or a thing. For example, one class of physical characteristics is biometric characteristics of a person. As used herein, the term biometric characteristics includes any measurable biological, physiological, or biomechanical characteristics of a person. Such characteristics include, for example, fingerprint, iris, DNA, typing patterns, voice, blood, etc. Thus, for computer security applications of the present invention, any biometric characteristic, or combination of biometric characteristics, which could reasonably identify a particular person would be appropriate. Another class of physical characteristics are the characteristics of things (e.g. non-biological items). Such characteristics include, for example, chemical or molecular composition, etc. Although the following description describes the invention in terms of parameters representing measurements of physical characteristics, it is to be understood that the invention is not limited to any particular type of parameters, but instead applies to any type of varying parameters.

Actual measurement techniques will vary depending on the embodiment of the invention, and the details of particular measurement techniques are not the subject of this invention. It is the use of the physical measurements, once obtained, to generate a repeatable cryptographic key for use in computer security applications that is the subject of this invention. Thus, we assume that the measurements have been made and that we are supplied with physical measurement parameters, represented here as parameters: $\phi_1, \phi_2, \cdots \phi_m$.

It is expected that the measurements of the same physical characteristics of the same entity may vary from one measurement to the next. This may be due to two factors. First, the measured physical characteristic itself may be dynamic. For example, if the measured physical characteristic were the biometric characteristics of voice (e.g. pitch, amplitude) during the speaking of some word or phrase, it is expected that an individual's voice characteristics will not be exactly the same during repeated vocalizations of the same word or phrase. As such, the measured characteristics represented as parameters $\phi_1, \phi_2, \cdots \phi_m$ will be different even though the same person speaks the same word or phrase. Second, the measured physical characteristic itself may be static, but the measurement of the physical characteristic may be imprecise or incomplete. For example, a person's fingerprint is static, but the results of different measurements of the same fingerprint may change slightly, due to imprecise measuring equipment, or because of an incomplete sample.

One of the advantages of the present invention is its ability to generate a repeatable cryptographic key in view of varying measurements. As a result, the invention may be used for various computer security applications. For example, as will be described in further detail below, in one embodiment the inventive technique is used to authenticate computer system users based on particular keystroke features during entry of a password. A user must be authenticated even though his/her keystroke features will change slightly during multiple entries of the same password. The present invention will generate a repeatable key for a particular user if the keystroke feature measurements vary within certain tolerances, thus authenticating the user.

In another embodiment (also discussed in further detail below) the invention may be used to protect information in a sensitive database containing private information. For example, consider a database which contains private information of convicted felons which is only meant to be used for legitimate criminal investigation purposes. For example, if a criminal is suspected of a crime the database record for only that particular criminal should be accessible by investigators. As such, each record in the database is encrypted and may only be decrypted using a key which is generated using DNA measurements of the associated person. This is to assure that prior to accessing potentially private information about a person, the person's DNA has already been recovered from a crime scene, thus making the person a suspect in a crime. While a person's DNA is not dynamic, imprecise measurement techniques or imprecise samples may result in having incomplete DNA measurements. In accordance with the techniques of the present invention, a repeatable key to unlock the database records could be generated with this incomplete DNA information.

In advantageous embodiments, the invention may be implemented using a programmable computer. Such a computer comprises a processor for executing computer program code stored in a memory which is accessible by the processor. As used herein, the term memory is used to refer to any computer readable medium, including without limitation, random access memory (RAM), read only memory (ROM), magnetic disk, optical disk, and holographic memory. In an advantageous embodiment, the computer program code is stored in a high speed memory, such as a RAM, which is connected to the computer processor. The computer program code required to implement the invention may be written in any well known computer language. Given the present description of the invention, one of ordinary skill in the art to which the invention pertains could readily write the program code necessary to implement the invention. The data structures described below (e.g. the share table) are also stored in memory. A user interacts with the computer using well known input/output devices and techniques (e.g. display screen, keyboard, mouse). Programmable computers of the type described herein are well known in the art and as such, further details are not required here.

As described above, we will assume that measurements of some physical characteristic have been taken and we are thus in possession of parameters $\phi_1, \phi_2, \cdots \phi_{m'}$ representing those physical characteristics. In one embodiment of the invention, a function *f* is applied to the parameters $\phi_1, \phi_2, \cdots \phi_{m'}$ in order to generate a set of indices $\psi_1, \psi_2, \cdots \psi_m$ such that $f(\phi_1, \phi_2, \cdots \phi_{m'}) = \{\psi_1, \psi_2, \cdots \psi_m\}$. (The subscript m' is used for the parameters (ϕ) to indicate that there need not be the same number of parameters (ϕ) and indices (ψ)). The indices $\psi_1, \psi_2, \cdots \psi_m$ are used to access a set of stored values, where at least some of the stored values are so-called cryptographic shares of a secret sharing scheme. Secret sharing schemes and cryptographic shares will be described in further detail below. At this point, let it suffice to say that possession of a sufficient number of valid cryptographic shares of a secret sharing scheme may be used to generate the repeatable key. In order to provide security, not all the stored values represent valid cryptographic shares. Instead, the particular values in the set of values which contain valid cryptographic shares are chosen to correspond to values of the indices $\psi_1, \psi_2, \cdots \psi_m$ which are expected to be generated during a legitimate attempt to access the computer resource. The other values in the set (i.e., those for which corresponding indices are not expected to be generated during a legitimate access attempt) do not contain valid cryptographic shares. As will be described in further detail below, a designer of the security system will be able to identify the indices which would be expected to be generated during a legitimate access attempt.

As an example, consider the physical measurement parameters $\phi_1, \phi_2, \phi_3, \phi_4$ corresponding to 4 measured keystroke features of a particular user during the typing of a password. Further, assume that the function f, when applied to these parameters, produces a set of indices, each in the range of 1-10. For example, the indices in the range of 1-10 may be a result of normalizing the latency measurements. Also, assume that the user has engaged in a training session during which the user typed the password five times. After applying the function f to the parameters $\phi_1, \phi_2, \phi_3, \phi_4$ generated during the training session, the indices $\psi_1, \psi_2, \psi_3, \psi_4$ were generated as follows:

password entry 1: $\psi_1, \psi_2, \psi_3, \psi_4 = 1,5,9,3$
password entry 2: $\psi_1, \psi_2, \psi_3, \psi_4 = 2,6,9,3$
password entry 3: $\psi_1, \psi_2, \psi_3, \psi_4 = 2,5,8,4$
password entry 4: $\psi_1, \psi_2, \psi_3, \psi_4 = 3,5,8,3$
password entry 5: $\psi_1, \psi_2, \psi_3, \psi_4 = 1,6,7,3$

From this information, a security designer can conclude that expected values of the indices for this user are as follows:

INDEX	EXPECTED VALUES
ψ_1	1,2,3
ψ_2	5,6
ψ_3	7,8,9
ψ_4	3,4

The indices are used to select values from a data structure called a share table, so named because it contains cryptographic shares. An exemplary share table is shown in Fig. 1 as table 100. The indices select values from the share table as follows. The first index ψ_1 selects a value from the first row, the second index ψ_2 selects a value from the second row, the third index ψ_3 selects a value from the third row, and the fourth index ψ_4 selects a value from the fourth row. The value of the particular index determines the column from which the value is selected. Thus, using conventional $(row,column)$ notation, given an index ψ_i a value is selected from the share table 100 at location (i,ψ_i).

Based on the expected values found during the training session, the locations in the share table 100 corresponding to the expected values for each of the indices will be set to contain valid cryptographic shares such that the repeatable key may be generated from those valid cryptographic shares. Those locations which do not correspond to the expected values for each of the indices will be set to contain values which are not valid cryptographic shares, called invalid cryptographic shares, such that the repeatable key may not be generated from those values. In Fig. 1, locations in share table 100 which contain valid cryptographic shares are identified with a ✓ and locations which contain invalid cryptographic shares are identified with a ×.

Secret sharing schemes will be described in further detail below. For purposes of the present example, assume that four valid cryptographic shares are required to generate the repeatable key. Also assume that on a particular access attempt the user types his/her password and the function *f* generates the following indices based on measurements of the keystroke features:

access attempt 1: $\psi_1, \psi_2, \psi_3, \psi_4 = 2,6,8,4$

Note that although this exact set of indices did not appear in any of the training sessions, each of the values is an expected value for the corresponding index. As such, the indices correspond to locations in the share table 100 which contain valid cryptographic shares. More particularly, index ψ_1 selects row 1, column 2 from the share table, index ψ_2 selects row 2, column 6 from the share table, index ψ_3 selects row 3, column 8 from the share table, and index ψ_4 selects row 4, column 4 from the share table. As can be seen from Fig. 1, each of these share table locations contains a valid cryptographic share as indicated by a ✓. As such, the correct repeatable key could be generated from the valid cryptographic shares and the user would be authenticated.

Alternatively, assume that the user types his/her password and the function f generates the following indices based on measurements of the keystroke features:

access attempt 2: $\psi_1, \psi_2, \psi_3, \psi_4 = 2,7,5,4$

Note here that the index values for ψ_2, ψ_3 are not expected values for these indices and as such, the entries in the share table 100 corresponding to these indices (row 2, column 7 and row 3, column 5) contain × indicating an invalid cryptographic share. As such, the correct repeatable key could not be generated from these cryptographic shares and the user would not be authenticated. Further details of an advantageous keystroke feature authentication embodiment of the invention will be given below.

According to one embodiment of the invention, errors, or variations, in measured physical characteristics may also be tolerated by selecting share table locations which were not actually indexed by the set of indices generated by the function f. This error tolerance is advantageously accomplished by selecting share table locations which are in the vicinity of locations actually indexed by the indices generated by the function f. For example, assume the share table 100 shown in Fig. 1 and an embodiment in which four indices $\psi_1, \psi_2, \psi_3, \psi_4$ are generated. If the repeatable key is not generated correctly by selecting shares from the share table locations indexed by $\psi_1, \psi_2, \psi_3, \psi_4$, then the system may first attempt to vary the selection of a share table location associated with the first index ψ_1 by choosing, during two separate key generation attempts, the share table location on either side of the location indexed by ψ_1. If the generation of the repeatable key is unsuccessful, the system may then attempt to vary the selection of a share table location associated with the second index ψ_2 in a similar manner. These steps of varying the actual chosen share table location may continue for each of the indices. Of course, this process of varying the selection of share table locations may be extended in order to implement an appropriate security scheme for different embodiments of the invention. For example, during each key generation attempt, one or more share table locations may be chosen using this variation technique. Further, the amount of variation allowed for each location may vary. As described above, the variation consists of one table location on either side of the actual indexed location. However, the variation may also consist of more than one table location on either side of the actual indexed location. Additionally, depending on the implementation, it may be appropriate to vary the selection of a share table location for only certain of the indices. Of course, many other variation techniques may be used and are contemplated by this description.

Advantageously the invalid cryptographic shares in a share table should be chosen such that they cannot be readily recognized as invalid by an adversary who compromises the computer system data and gains access to the share table. In this way, the adversary would not be able to gain any information from the share table. Further, the values in a share table may be encrypted for further security. For example, as will be described in further detail below, in the keystroke feature authentication embodiment, the values in the share table are encrypted with the user's actual password in order to provide additional security.

Thus, as can be seen, the techniques in accordance with the present invention allow a repeatable key to be generated notwithstanding variations in measured physical characteristics. The amount of variation allowed is defined by the configuration of the share table as well as the error tolerance technique described above.

As described above, a share table contains at least some valid cryptographic shares of a secret sharing scheme. There are various types of secret sharing schemes, and various ones of these schemes may be used to generate the cryptographic shares for storage in the share table. One type of secret sharing scheme is a polynomial secret sharing scheme, the details of which are described in A. Shamir, *How To Share A Secret*, Communications of the ACM 22(11):612-613, November 1979, which is incorporated herein by reference. Since a detailed description of the scheme may be found in the reference, the scheme will only be described in general terms herein. A secret key k is chosen from the integers *mod* q, where q is prime. (For notational simplicity, unless otherwise stated, the calculations described below for generating the secret key k are performed *mod* q). Then, a threshold t is determined which represents how many cryptographic shares are required to generate the secret key k. This is a design choice that would be determined based on the particular embodiment of the invention. Next a degree t-1 polynomial p is chosen at random such that $p(0) = k$. This polynomial is illustrated in Fig. 2. Note that k is illustrated as the point at which the polynomial crosses the y axis, that is, $p(0) = k$. As is well known, a polynomial of degree d can be determined from a knowledge of d+1 points on that polynomial by using well known interpolation techniques. Thus, since p is a degree t-1 polynomial, knowledge of t points on that polynomial are sufficient to determine p. Further, once p is known, $p(0)$ can be determined, which is the secret key k. Thus, by knowing t points on the polynomial, the secret key k can be determined as follows.

Given points $(x_1, p(x_1)), \ldots, (x_t, p(x_t))$ on the degree-(t-1) polynomial p over the integers mod q, the value $k = p(0)$ can be computed as

$$p(0) = \sum_{i=1}^{t} c_i p(x_i)$$

where

$$c_i = \prod_{1 \le j \le t, j \ne i} \frac{x_j}{x_j - x_i}$$

Thus, in accordance with one embodiment of the invention, the valid cryptographic shares which are stored in the share table are points on the polynomial p. Any number of polynomial points can be stored in the table, with an understanding that access to any t of those points will be enough to generate the secret key k.

Referring to Fig. 2, examples of points on the polynomial, and therefore valid cryptographic shares in the polynomial secret sharing scheme, would be $(x_1, p(x_1)), (x_2, p(x_2)), (x_3, p(x_3)), (x_4, p(x_4))$. These valid cryptographic shares would be stored in share table locations which, as described above, correspond to expected index values which would be generated during a legitimate access attempt. Invalid cryptographic shares, to be stored in share table locations which do not correspond to expected index values, can be chosen from points off the polynomial p, for example points 201, 202, 203.

Thus, in a straightforward approach, polynomial points are stored in the share table locations as pairs $(x_i, p(x_i))$.

In an alternative to the straightforward approach, the cryptographic shares may be compressed in order to save memory space. In this approach, the inherent sequential ordering of the share table locations is utilized and points on the polynomial are chosen accordingly. Consider the share table 300 shown in Fig. 3. The share table 300 has 16 locations. In the upper left corner of each location is shown the standard *(row,column)* index of the location. In the upper right corner of each location is shown the sequence number of the location. In a table having n columns, the standard *(row,column)* index can be converted to the sequence number by the equation: $\text{sequence}\ \# = ((row - 1) \cdot n) + column$. For example, in the table shown in Fig. 3, $n = 4$, and table entry (3,2) has a sequence number of $((3-1) \cdot 4) + 2 = 10$. In this embodiment, a valid cryptographic share is entered into a table location having sequence number s_i by placing in that location the value $p(s_i)$. Thus, only the y value of the polynomial point is stored in the table. The x value of the polynomial point is the sequence number of the share table location and can be derived from the standard *(row,column)* index. For example, in Fig. 3, location indexed by (3,2) would contain the value $p(10)$. It is noted that in this embodiment, prior to generating a valid cryptographic share for storage in the share table, the particular location within the share table in which the cryptographic share will be stored must be known. The cryptographic share is decompressed using the stored *y* value in conjunction with the location in which it is stored. In this embodiment, the value stored in memory is considered a cryptographic share, and is stored in compressed format.

We now describe an embodiment which stores encrypted shares in the share table. In accordance with this embodiment, the share table is stored as a plurality of rows (corresponding to the number of parameters/indices that may be generated) but a single column. A share table 400 in accordance with this embodiment of the invention is illustrated in Fig. 4. In a manner similar to that described above in conjunction with Fig. 1, each row of the share table 400 (Fig. 4) corresponds to one of the indices ψ generated by applying the function *f* to a parameter ϕ. Recall, as described above, $f(\phi_1,\phi_2,\cdots\phi_{m'})=\{\psi_1,\psi_2,\cdots\psi_m\}$. The share table 400 stores cryptographic shares (e.g. points on the polynomial) which have been encrypted (by an encryption function *E*) using the expected values of the indices $\{\psi_1,\psi_2,\cdots\psi_m\}$ as the encryption key, such that a polynomial point (x_i,y_i) is encrypted as $E_{\psi_i\text{-expected}}(x_i,y_i)$. (The particular encryption function *E* would depend on the particular implementation.) As such, the stored encrypted shares may only be decrypted when an expected index value is generated. For example, the value stored in the share table 400 location corresponding to ψ_1 is $E_{\psi_1\text{-expected}}(x_1,y_1)$, which is a polynomial point (x_1,y_1) which has been encrypted using the expected value of ψ_1 as the encryption key. The polynomial point (x_1,y_1) can only be decrypted using the expected value of ψ_1 as the decryption key. Of course, when an unexpected value for an index is used to decrypt an encrypted share, the decrypted share will not be a valid polynomial point. Thus, in accordance with this embodiment, the share table will accommodate any number of values which may be generated for an index ψ. Any index value which is not expected will not be able to decrypt the stored encrypted share.

It is further noted that in accordance with the above described embodiment, a certain amount of parameter variation may be tolerated by varying the degree of the polynomial from which points are chosen as shares. As described above in conjunction with Fig. 2, given a degree t-1 polynomial, the knowledge of t points on that polynomial are sufficient to determine p and thus the secret key k. So, in this embodiment, even though an attempt may be made to decrypt m cryptographic shares, the degree of polynomial p may be chosen such that p may be determined from less than m points on the polynomial. Thus, even if one (or more) of the indices ψ is generated incorrectly, and as a result the decryption of the corresponding share results in a point which is not on the polynomial, the correct polynomial may still be determinable using other generated points which are on the polynomial. In this case, in which we possess a sufficient number of valid points on the polynomial, together with some invalid points off the polynomial, well known techniques may be used to exclude the bad points in the determination of p. For example, various subsets of the generated points may be used in multiple attempts to determine p, until some pattern emerges which indicates the valid and invalid points.

The above described embodiments were described assuming that the secret sharing scheme being used was a polynomial secret sharing scheme. However, other types of secret sharing schemes, some of which are described below, may also be used.

In another embodiment of the invention, the polynomial scheme is modified such that the secret key k is defined as $k = g^{p(0)} \bmod r$ where r is prime, r-1 is a multiple of q, and g has order q mod r, i.e., the set $\{g^0 \bmod r, g^1 \bmod r, g^2 \bmod r, \ldots\}$ contains q elements. Rather than storing $p(x_1), p(x_2), \ldots$ as cryptographic shares in the share table, instead the values $g^{p(x_1)} \bmod r, g^{p(x_2)} \bmod r, \ldots$ are stored as the cryptographic shares. Given pairs $(x_1, g^{p(x_1)} \bmod r), \ldots, (x_t, g^{p(x_t)} \bmod r)$, the value k can be computed as

$$k = \prod_{i=1}^{t} (g^{p(x_i)} \bmod r)^{c_i} \bmod r$$

where c_i is as defined above.

In another embodiment of the invention, the secret key k is computed as the determinant of am $m \times m$ matrix over the integers mod q. The cryptographic shares are vectors that are used to compose the matrix. For ease of description, we describe this embodiment for a table with two columns. In this embodiment, random column vectors $\vec{w}_1^0,...,\vec{w}_m^0$, each with m elements taken from the integers mod q, are selected such that $k = \left|\vec{w}_1^0,...,\vec{w}_m^0\right|$, i.e., k is the determinant of the matrix composed by these vectors. Another m random vectors $\vec{u}_1,...,\vec{u}_m$ are then computed so that

$$\forall b \in \{0,1\}^m : \left|\vec{u}_1^{b(1)},...,\vec{u}_m^{b(m)}\right| = 1$$

where $\vec{u}_i^{b(i)} = \vec{e}_i$ if $b(i) = 0$ and $\vec{u}_i^{b(i)} = \vec{u}_i$ if $b(i) = 1$. Here, $b(i)$ denotes the i-th bit of b, and $\vec{e}_i$ denotes the unit vector with a 1 in position i and 0 in all other positions. Then, vectors $\vec{w}_1^1,...,\vec{w}_m^1$ are computed as $\vec{w}_i^1 = \left(\vec{w}_1^0,...,\vec{w}_m^0\right) \cdot \vec{u}_i$. The vector $\vec{w}_i^0$ is the valid cryptographic share for the first column of the i-th row of the share table, and similarly $\vec{w}_i^1$ is the valid cryptographic share for the second column of the i-th row of the share table. It then follows that for any $b \in \{0,1\}^m$, $\left|\vec{w}_1^{b(1)},...,\vec{w}_m^{b(m)}\right| = k$. That is, with one valid cryptographic share from each row, the secret key k can be computed.

An efficient algorithm to generate $\vec{u}_1,...,\vec{u}_m$ is as follows. An upper-triangular matrix $U' = (u_1',...,u_m')$ is chosen that has 1 for each diagonal element and random integers mod q above the diagonal. Then, $(\vec{u}_1,...,\vec{u}_m) = \Pi \cdot U' \cdot \Pi^{-1}$ where $\Pi = (\vec{\pi}_1,...,\vec{\pi}_m)$ is a permutation matrix (i.e., the identity matrix with columns permuted).

We now describe a keystroke features authentication embodiment of the invention in which the inventive techniques are used to provide authentication in a computer system based on an entered password and keystroke features measured during entry of the password. In this embodiment, each user is associated with a password, identified as *pwd*. Further, when a user enters the *pwd*, a certain number, *m*, of features of the user's keystroke features are measured or derived. A measured feature is one that is actually measured. For example, if the *pwd* is 8 characters, in one illustrative embodiment there are 15 measured features, including 8 keypress durations and 7 latencies between keypresses. A derived feature is one which is not directly measured, but is derived from a measured feature (e.g., the ratio of the 7th and 8th keypress durations). Assume for purposes of this description that there are 15 features ($m = 15$). In this embodiment, a share table having *m* rows and 2 columns is used. Such a share table is shown in Fig. 5 and will described in further detail below.

Next a key *k* is chosen and cryptographic shares are generated using a secret sharing scheme as described above. Assume a polynomial secret sharing scheme is used and that the actual points of the polynomial are stored in the share table. Initially, 30 valid cryptographic shares are generated and each location in the share table 500 is initialized to contain a valid cryptographic share. In order to add an additional measure of security, all values in share table 500 are encrypted using *pwd* as the encryption key, and may be decrypted using *pwd* as the decryption key.

The function *f* which generates the indices ψ_i will now be described. For each measured keystroke feature parameter ϕ_i, there is an associated threshold h_i which is used during the generation of the index ψ_i for the particular measured feature. The function *f* is defined as follows:

$$f(\phi_1, \phi_2, \cdots \phi_m) = \{\psi_1, \psi_2, \cdots \psi_m\} \in \{0,1\}^m$$

where

$$\psi_i \begin{cases} 0 \text{ if } \phi_i < h_i \\ 1 \text{ if } \phi_i \geq h_i \end{cases}$$

Thus, for each parameter representing a measured keystroke feature, the corresponding index generated by the function will be 0 if the parameter is less than the associated threshold and 1 if the parameter is greater than or equal to the associated threshold. The threshold h_i may, for example, be a time value (e.g. 100ms).

Each time a user enters a *pwd* each of the 15 keystroke features will be measured as a parameter ϕ_i. The function *f* will compare the parameter ϕ_i to the threshold h_i

associated with that measurement and will generate ψ_i as having a value of 0 or 1 depending on which side of the threshold ϕ_i falls on. The value of ψ_i is then used as an index into the *i*th row of the table 500. The left column of row *i* is chosen if ψ_i is 0, and the right column of row *i* is chosen if ψ_i is 1. The value in the selected entry of the *i*th row is used as the cryptographic share from that particular row. After cryptographic shares have been selected from all 15 rows, they are used as described above to generate the key *k*. As described above, the cryptographic shares must first be decrypted using *pwd* prior to using them to generate the key *k*.

Since all share table 500 locations initially have valid cryptographic shares, initial access attempts which provide the correct *pwd* (to allow decryption of the share table 500) will result in the generation of the correct key *k*. However, during subsequent access attempts, the access structure (i.e., the sets of entries in the share table which enable generation of the cryptographic key) defined by share table 500 is reduced based on the typing patterns of the user. This is accomplished through use of a history table, which stores the parameters $\phi_1, \phi_2, \cdots \phi_m$ for a given number of previous successful access attempts for each user. An illustrative history table 600 is shown in Fig. 6. As the history table 600 is populated with parameters, certain typing patterns of the associated user can be determined. Each of the measured parameters can be analyzed to determine if one of the parameters is associated with a distinguishing feature. A distinguishing feature is one which can be used to distinguish this user from other users. If a parameter ϕ_i is consistently substantially more or less than the associated threshold h_i, then the associated feature is considered a distinguishing feature, and the share table 500 may be rebuilt to make use of this distinguishing feature as follows. If a parameter ϕ_i in the history table 600 is consistently on one side of the associated threshold h_i, then it is assumed that this parameter ϕ_i will continue to be on the same one side of the associated threshold h_i during subsequent legitimate access attempts by the same user. As such, the entry in the share table 500 corresponding to the other side of the threshold for this parameter will be changed to contain an invalid cryptographic share. Thus, the access structure defined by share table 500 is thus reduced and in this way the share table 500 is modified to reflect the typing patterns of the authorized user.

The history table can be analyzed and corresponding changes made to the share table 500 on a periodic basis. This may be done after every successful access by the user, after some number of successful accesses, or after some time period has elapsed. In an advantageous embodiment, when making changes to the share table 500, the key *k* remains the same, but the entire table is rebuilt using new cryptographic shares which are points on a new polynomial *p*. In this manner the share table, and therefore the access structure, dynamically adjust based on the user's typing patterns. Of course, depending on the typing patterns of the user, the share table may be modified to contain more, or less, valid cryptographic shares.

Also, as described above, the degree of the polynomial *p* may be chosen to allow generation of the correct cryptographic key even though one (or more) of the selected share table locations contain an invalid cryptographic share.

Thus, in order for the correct *k* to be generated, the user must enter the correct *pwd* (so that the cryptographic shares in the share table can be decrypted), and the *pwd* must be entered using keystroke features that will result in a sufficient number of valid cryptographic shares being chosen from the share table. A user could then be authenticated on the basis of whether the correct secret key *k* was generated. For example, this could be verified by decrypting a file with *k* and seeing if a correctly formatted file results, or by hashing *k* and comparing the result to a previously stored hash value.

We have found that it is advantageous to further implement an error tolerance scheme, as described above, in the keystroke features authentication embodiment. More particularly, we have found it advantageous to simultaneously vary the selection of two share table 500 locations in an attempt to generate the repeatable key. The variation consists of, for each of two share table rows, choosing the share from the column other than the column actually indexed by the associated index generated by the function *f*. Thus, if a repeatable key is not generated correctly based on the originally generated indices, then two rows of the share table 500 are chosen for variation. If the repeatable key is still not generated correctly, then another two rows are chosen for variation. This continues until the correct repeatable key is generated or until all two-row combinations have been exhausted.

The repeatable key *k* may also be used as an encryption key to encrypt the user's data in the computer.

We now describe an embodiment of the invention in which the inventive techniques are used to protect a database which stores private information of individuals (e.g., convicted felons) which is only meant to be used for legitimate criminal investigation purposes. For example, if a criminal is suspected of a crime the database record for only that particular criminal should be accessible by investigators. As such, each record in the database is encrypted and may only be decrypted using a key which is generated using DNA measurements of the associated person. This is to assure that prior to accessing potentially private information about a person, the person's DNA has already been recovered from a crime scene, thus making the person a suspect in a crime. While a person's DNA is not dynamic, imprecise measurement techniques or imprecise samples may result in having incomplete DNA measurements. In accordance with the techniques of the present invention, a repeatable key to unlock the database records could be

This embodiment of the invention utilizes the technique of storing shares generated with this incomplete DNA information.encrypted with expected index values, which was described above in conjunction with Fig. 4. When a record containing sensitive information is to be stored in the database, assume that certain DNA measurement of a person are known. Referring again to Fig. 4, assume that *m* DNA measurements have been taken. An appropriate polynomial *p* is chosen of degree *t-1*, where it is desired that knowledge of *t* points on the polynomial will allow access to the record, where $t \leq m$. The record is encrypted using a key *k* which is the point on the polynomial *p* which crosses the *y* axis, and the encrypted record is stored in the database. A share table which would allow the record to be decrypted in appropriate circumstances is generated as follows. Each DNA measurement is assigned a sequence number (1, 2, … m) which correspond to parameters $\phi_1, \phi_2, \cdots \phi_m$ representing those measurements. A function *f* is chosen such that $f(\phi_1, \phi_2, \cdots \phi_{m'}) = \{\psi_1, \psi_2, \cdots \psi_m\}$. For example, the function *f* may be a hash function. Using knowledge of the expected values of $\psi_1, \psi_2, \cdots \psi_m$ together with knowledge of the polynomial *p*, points on the polynomial *p* are encrypted using expected index values as described above in conjunction with Fig. 4.

Thereafter, the stored record of a person may only be decrypted by someone in possession of the DNA of that person. In order to decrypt the record, the appropriate sequence of DNA measurements are made in order to generate the parameters $\phi_1, \phi_2, \cdots \phi_{m'}$ representing those measurements. The function *f* is applied to the parameters $\phi_1, \phi_2, \cdots \phi_{m'}$ in order to generate indices $\psi_1, \psi_2, \cdots \psi_m$ such that $f(\phi_1, \phi_2, \cdots \phi_{m'}) = \{\psi_1, \psi_2, \cdots \psi_m\}$. Each generated index ψ_i is used to decrypt the corresponding encrypted share ($E_{\psi_i\text{-expected}}(x_i, y_i)$) stored in the *i*th row of the share table in order to generate the polynomial point (x_i, y_i). The set of generated polynomial points may be used as described above in order to generate the cryptographic key *k*, which is the point at which the polynomial crosses the y axis.

The foregoing Detailed Description is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the patent laws. It is to be understood that the embodiments shown and described herein are only illustrative of the principles of the present invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.